REMARKS

Status of Claims:

Claims 1-36 and 49-85 are present for examination. Claims 37-48 are withdrawn.

Specification:

The specification has been amended to correct a minor informality.

Indefiniteness Rejection:

Claims 1-36 and 49-85 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

With respect to claims 1-36 and 49-85, as amended, the rejection is respectfully traversed.

The Examiner stated that for claims 1, 12, 25, 49, 60, and 73, "the wherein clause is conclusory and it is not clear what or how the possibility of presence of dependence is detected when present in fact and detect dependence when not in fact." Claims 1, 12, 25, 49, 60, and 73 have been amended to recite a "means for detecting" as a means by which the possibility of presence of a dependence is detected. Therefore, claims 1, 12, 25, 49, 60, and 73 are now considered to be in condition for allowance.

The Examiner stated that in claims 1, 12, 25, 49, 60, and 73, "it is not clear what is meant by 'allow to detect said dependence in fact', since there is no dependence to detect." Claims 1, 12, 25, 49, 60, and 73 have been amended to allow the detector to detect a "pseudo presence" of a dependence if there is no dependence in fact. Support for such language may be found in the originally filed application at page 76, line 4. Therefore, claims 1, 12, 25, 49, 60, and 73 are now considered to be in condition for allowance.

Prior Art Rejection:

Claims 1-36 and 49-85 are rejected under 35 U.S.C. 102(b) as being anticipated by Kiyohara et al. (U.S. Patent 5,694,577) (hereinafter Kiyohara).

With respect to claims 1-36 and 49-85, as amended, the rejection is respectfully traversed.

Independent claim 1, as amended, recites a detector for detecting at least one kind of address dependence between a first instruction and a second subsequent instruction executed by at least a processor, said detector comprising:

"means for, if said at least one kind of address dependence is present in fact, detecting said presence in fact of said at least one kind of address dependence, and if said at least one kind of address dependence is not present in fact, detecting a pseudo presence of said at least one kind of address dependence, and

means for causing a recovery process to be executed by said processor when said processor executes said second instruction and either said means for detecting detects said presence in fact of said at least one kind of address dependence or said means for detecting detects said pseudo presence of said at least one kind of address dependence." (Emphasis Added)

The detector including the above-quoted features has the advantage that the detector can cause a recovery process to be executed by a processor when the processor executes an instruction that is checked for a dependence and there is either a presence in fact of a dependence or a pseudo presence of a dependence. Thus, there is no need for an additional instruction to be placed in the code to cause the recovery process to be executed, because the recovery process is caused to be executed when the processor executes an instruction that is checked for a dependence. By not requiring an additional instruction to cause the recovery process, the code length is not increased to include such an instruction, and the hardware does not need to support an additional opcode for the new instruction. (page 66, lines 4-22).

The detector could cause a recovery process to be executed without an additional instruction, for example, by having as input a speculative execution flag and as output a data dependence detected result. (figure 3, reference numbers 10, 13, 16). When the processor enters a speculative execution state, it would set the speculative execution flag. (figure 4, reference "speculative execution flag"; page 63, lines 10-15). Thereafter, if an instruction is checked for a dependence and there is detected a presence in fact of a dependence or a pseudo presence of a dependence, the data dependence detected result would be set and the processor would be caused to start a recovery process. (figure 4, reference "data dependence detected result"; page 66, lines 4-22).

Kiyohara neither discloses nor suggests the detector including the above-quoted features where a recovery process is caused to be executed by a processor when an instruction is checked for a dependence and either a presence in fact of a dependence or a pseudo presence of a dependence is detected. The Memory Conflict Buffer (MCB) scheme in Kiyohara requires the introduction of two new instructions: (1) preload; and (2) check, which directs the hardware to determine if a dependence violation has occurred and to branch to conflict correction code as required. (Kiyohara; column 2, lines 23-29). Thus, the code that is executed by the processor in Kiyohara must be changed by the compiler to include the additional instructions. (Kiyohara; figure 2(a), 2(b); column 2, lines 29-42). The addition to the code, in Kiyohara, of a potentially large number of check instructions results in execution overhead and code expansion. (Kiyohara; column 11, lines 18-23).

Furthermore, the recovery process in Kiyohara is not caused to be executed when an instruction that is checked for a dependence is executed, but rather when an inserted "check" instruction is executed. (Kiyohara; column 5, lines 28-36). Thus, the device in Kiyohara does not satisfy the requirement of the present claim that a recovery process is caused to be executed when the processor executes a second instruction that is checked for a dependence.

Therefore, independent claim 1 is neither disclosed nor suggested by the cited prior art and, hence, is believed to be allowable.

Independent claim 12, as amended, recites a detector provided in a self-processor included in a multiple processor system including said self-processor and at least a processor other than said self-processor, and said multiple processor system performing parallel processings in thread units of program, and said detector detecting at least one kind of dependence in address between an instruction included in a thread executed by said self-processor and an instruction included in a thread executed by said other processor, said detector being adopted to detect a possibility of presence of said at least one kind of dependence, said detector comprising:

"means for, if said at least one kind of dependence in address between an instruction included in a thread executed by said self-processor and an instruction included in a thread executed by said other processor is present in fact, detecting said presence in fact of said at least one kind of dependence, and if said at least one kind of dependence in address between an instruction included in a thread executed by said self-processor and an instruction included in a thread executed by said other processor is not present in fact, detecting a pseudo presence of said at least one kind of dependence." (Emphasis Added)

The detector provided in a self-processor included in a multiple processor system including the above-quoted features has the advantage that the detector can detect a dependence or pseudo presence of a dependence between an instruction executed in a thread by the self-processor and an instruction executed in a thread by another processor. (page 118, line 21 to page 120, line 19).

Kiyohara neither discloses nor suggests the detector including the above-quoted features where the detector can detect a dependence or pseudo presence of a dependence between an instruction executed in a thread by a self-processor and an instruction executed in a thread by another processor. In Kiyohara, there is only disclosed a single processor system and, thus, there is no teaching of how to execute threads in a multiple-processor system and how to detect dependencies between instructions executed in threads by two different processors in a multiple-processor system. (Kiyohara; figure 3; column 1, lines 14-22).

Therefore, independent claim 12 is neither disclosed nor suggested by the cited prior art and, hence, is believed to be allowable.

Independent claim 25 recites a detector provided in a self-processor included in a multiple processor system similar to the detector provided in a self-processor included in a multiple processor system of claim 12. Therefore, independent claim 25 is believed to be allowable for at least the same reasons claim 12 is believed to be allowable.

Independent claim 49 recites a semiconductor integrated circuit including a detector similar to the detector of claim 1 and, thus, is believed to be allowable for at least the same reasons claim 1 is believed to be allowable.

Independent claim 60 recites a semiconductor integrated circuit including a detector provided in a self-processor included in a multiple processor system similar to the detector provided in a self-processor included in a multiple processor system of claim 12. Therefore, independent claim 60 is believed to be allowable for at least the same reasons claim 12 is believed to be allowable.

Independent claim 73 recites a semiconductor integrated circuit including a detector provided in a self-processor included in a multiple processor system similar to the detector provided in a self-processor included in a multiple processor system of claim 12. Therefore, independent claim 73 is believed to be allowable for at least the same reasons claim 12 is believed to be allowable.

The dependent claims are deemed allowable for at least the same reasons indicated above with regard to the independent claims from which they depend.

Conclusions:

Applicant believes that the present application is now in condition for allowance. Favorable reconsideration of the application as amended is respectfully requested.

The Examiner is invited to contact the undersigned by telephone if it is felt that a telephone interview would advance the prosecution of the present application.

The Commissioner is hereby authorized to charge any additional fees which may be required regarding this application under 37 C.F.R. §§ 1.16-1.17, or credit any overpayment, to Deposit Account No. 19-0741. Should no proper payment be enclosed herewith, as by a check being in the wrong amount, unsigned, post-dated, otherwise improper or informal or even entirely missing, the Commissioner is authorized to charge the unpaid amount to Deposit Account No. 19-0741. If any extensions of time are needed for timely acceptance of papers submitted herewith, Applicant hereby petitions for such extension under 37 C.F.R. §1.136 and authorizes payment of any such extensions fees to Deposit Account No. 19-0741.

Respectfully submitted,

Date 4 November 2004

By [signature] 36,489

FOLEY & LARDNER LLP
Customer Number: 23392
Telephone: (310) 975-7895
Facsimile: (310) 557-8475

David A. Blumenthal
Attorney for Applicant
Registration No. 26,257